LINQ

(Language Integrated Query)

OLD Days (Loops with If statements)

No easy way to query collections of objects.

```
List<Employee> nashvilleEmployees = new List<Employee>();
foreach (var emp in employees)
{
    if (emp.City == "Nashville")
    {
        nashvilleEmployees.Add(emp);
    };
}
foreach (var employee in nashvilleEmployees)
{
    Console.Write("NASHVILLE EMPLOYEES:  " +employee.Name + " " + employee.City);
}
```

OLD Days (Embedded Sql)

```
StringBuilder sb = new StringBuilder();
sb.Append("Select ani.CommonName,");
sb.Append("ani.Gender, ani.Species,");
sb.Append("ani.AcquiredDate,");
sb.Append("h.Name, h.[Description] ");
sb.Append("from Animals ani join Habitat h ");
sb.Append("on ani.HabitatId = h.HabitatId ");
sb.Append("order by ani.CommonName");
var query = sb.ToString();
using (SqlConnection connection = new SqlConnection("Data Source=alienware-pc\\sqlexpress;Initial C
using (SqlCommand cmd = new SqlCommand(query, connection))
{
    connection.Open();
    using (SqlDataReader reader = cmd.ExecuteReader())
    {
        // Check is the reader has any rows at all before starting to read.
        if (reader.HasRows)
        {
            while (reader.Read())
            {
                Console.WriteLine("Animal Name: " + reader[0] + "  Habitat: " + reader[4]);
            }
        }
        Console.ReadLine();
    }
```

Code Base Complications (Old Days)

Tons of looping and if statements.

Making loops hit database could be bad so had to do work arounds

Hard to debug embedded sql. Hard to maintain too.

Hard to work with objects and sql result in combination.

LINQ to the Res

Introduced in Visual Studio 2008 and designed by Anders Hejlsberg, LINQ (Language Integrated Query) allows writing queries even without the knowledge of query languages like SQL, XML etc. LINQ queries can be written for diverse data types. (objects, collections, xml, databases…)

Example of a LINQ query

C#

```
using System;
using System.Linq;

class Program
{
   static void Main()
   {
      string[] words = {"hello", "wonderful", "LINQ", "beautiful", "world"};

      //Get only short words
      var shortWords = from word in words where word.Length <= 5 select word;

      //Print each word out

      foreach (var word in shortWords)
      {
         Console.WriteLine(word);
      }

      Console.ReadLine();
   }
}
```

Comes in Two Flavors

Syntax of LINQ

There are two syntaxes of LINQ. These are the following ones.

Lamda (Method) Syntax

```
var longWords = words.Where( w => w.length > 10);
Dim longWords = words.Where(Function(w) w.length > 10)
```

Query (Comprehension) Syntax

```
var longwords = from w in words where w.length > 10;
Dim longwords = from w in words where w.length > 10
```

Query a collection of List<Objects>

```
List<SinglePassHolder> singleCustomersHavingActivePasses = singlePassHolders
    .Where(a => a.IsPassActive == false)
    .OrderBy(a => a.LastName).ToList();


Console.WriteLine("LIST OF INACTIVE SINGLE ZOO PASSHOLDERS NEEDING EMAIL REMINDER TO RENEW");

foreach (var customer in singleCustomersHavingActivePasses)
{
    var currentPass = customer.IsPassActive ? "Active" : "Inactive";

    Console.WriteLine("CustomerId: " + customer.CustomerId.ToString() + "  Name: " + customer.FirstName +
    Console.ReadLine();
}
```

Query Database Entities

```
//join data from two tables
var employeeDetails = (from emp in dbContext.Employee
                       join dept in dbContext.Department
                       on emp.DepartmentId equals dept.DepartmentId
                       orderby dept.Name
                       select new   //roll join results into a new object
                       {
                           Name = emp.Name,
                           Description = emp.Description,
                           DepartmentName = dept.Name,
                           SupervisorName=dept.SupervisorTitle
                       }).ToList();

foreach (var details in employeeDetails)
{
    Console.Write("Employee Name: " + details.Name + " Department Name: " + details.DepartmentName + "
    Console.WriteLine(Environment.NewLine);
}

Console.ReadLine();
```

Exercise

1. Run the following sql statement to create a sql expression version of the chinook database. File name is “Chinook_sqlServer.sql”.
2. Download and run this project: https://github.com/dtinsley333/LinqPractice
3. Modify the app to query at least 3 of the tables.
4. Write 5 linq queries that mirror 5 of the queries you did in the chinook sql practice. (orderby, take(), joins, group by…)